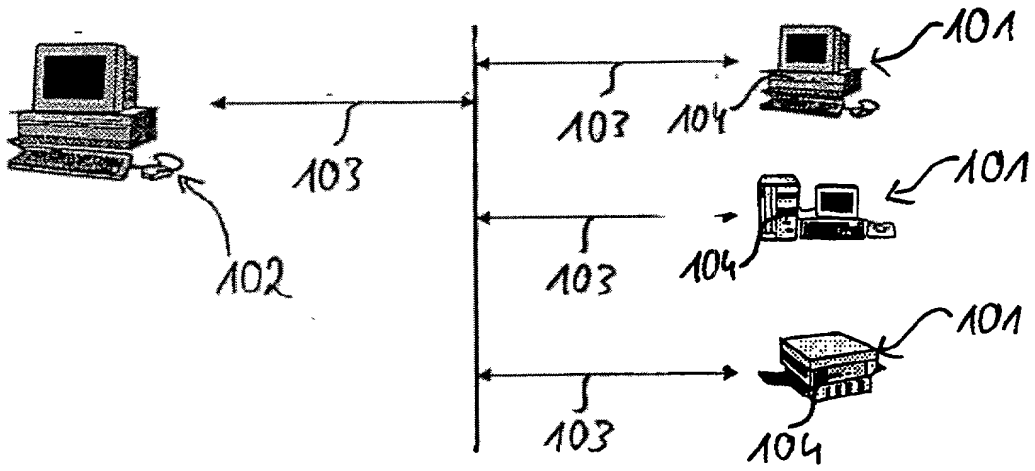


Related Art

Fig. 1A

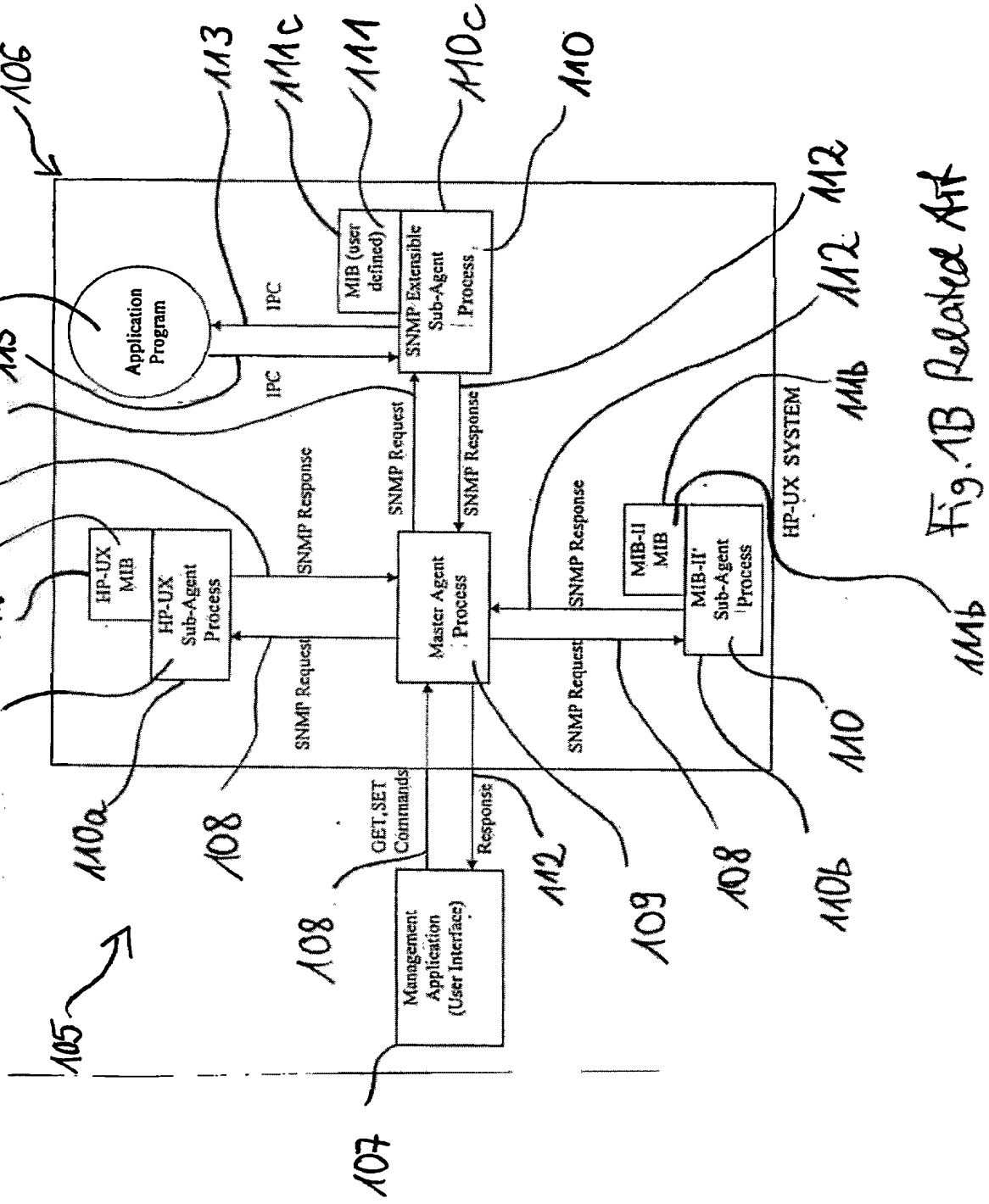


Fig. 1B Related Art

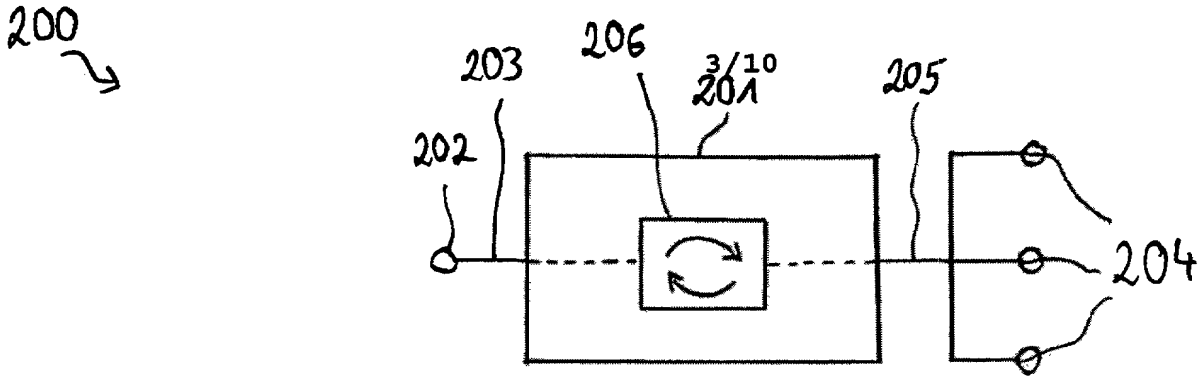


Fig. 2A

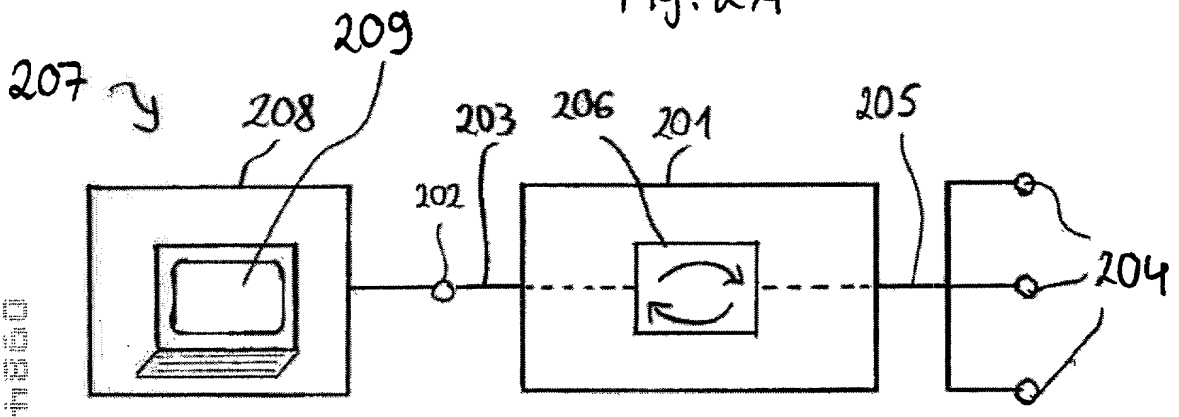


Fig. 2B

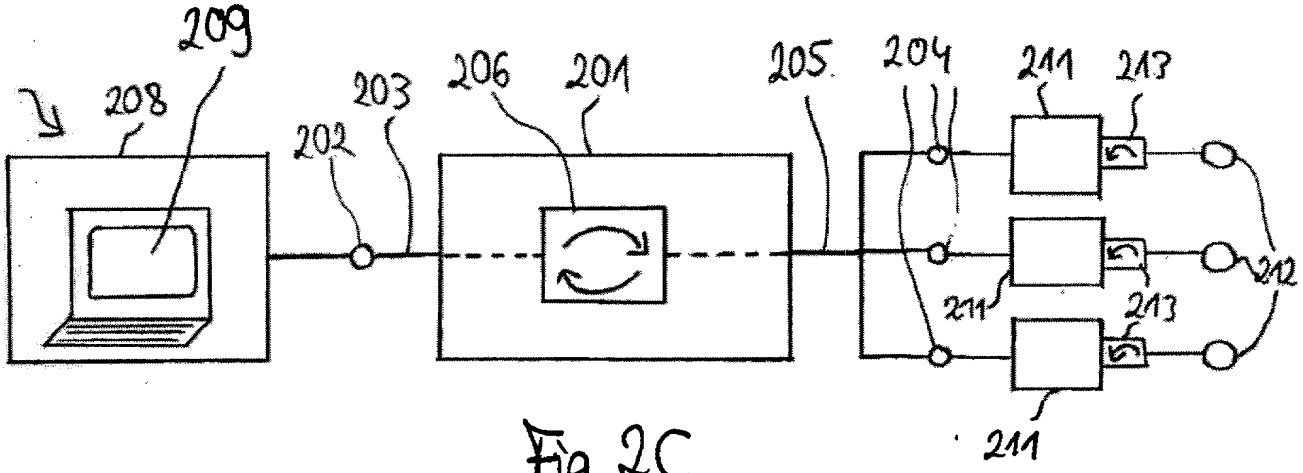


Fig. 2C

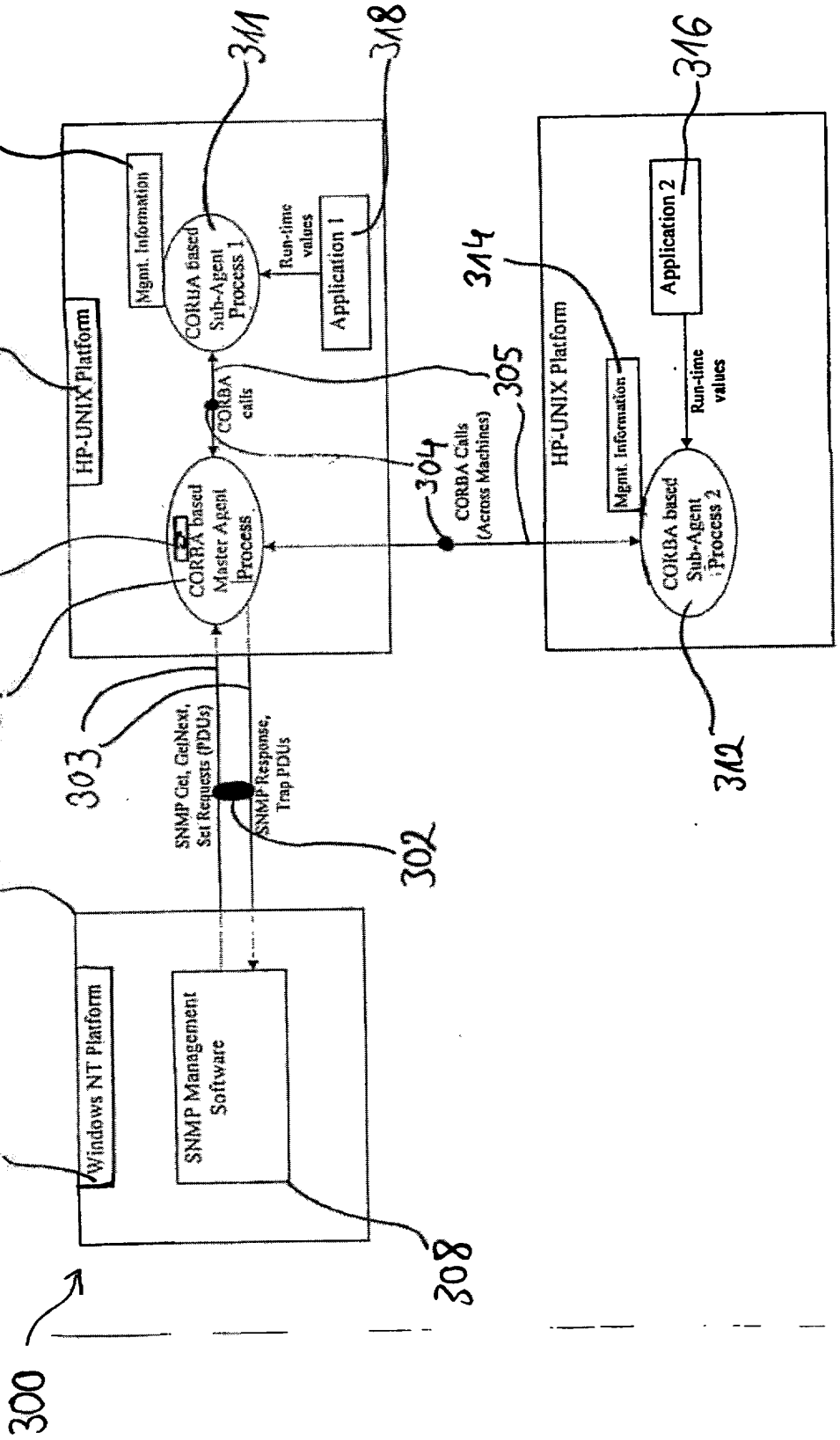

300
309
Windows NT Platform
SNMP Management Software
308
SNMP Get, GetNext, Set Requests (PDUs)
SNMP Response, Trap PDUs
302
303
301
306
310
HP-UNIX Platform
Mgmt. Information
313
CORBA based Sub-Agent Process 1
311
Run-time values
Application 1
318
CORBA calls
305
CORBA based Master Agent Process
304
CORBA Calls (Across Machines)
314
HP-UNIX Platform
Mgmt. Information
CORBA based Sub-Agent Process 2
312
Application 2
316
Run-time values

Fig. 3

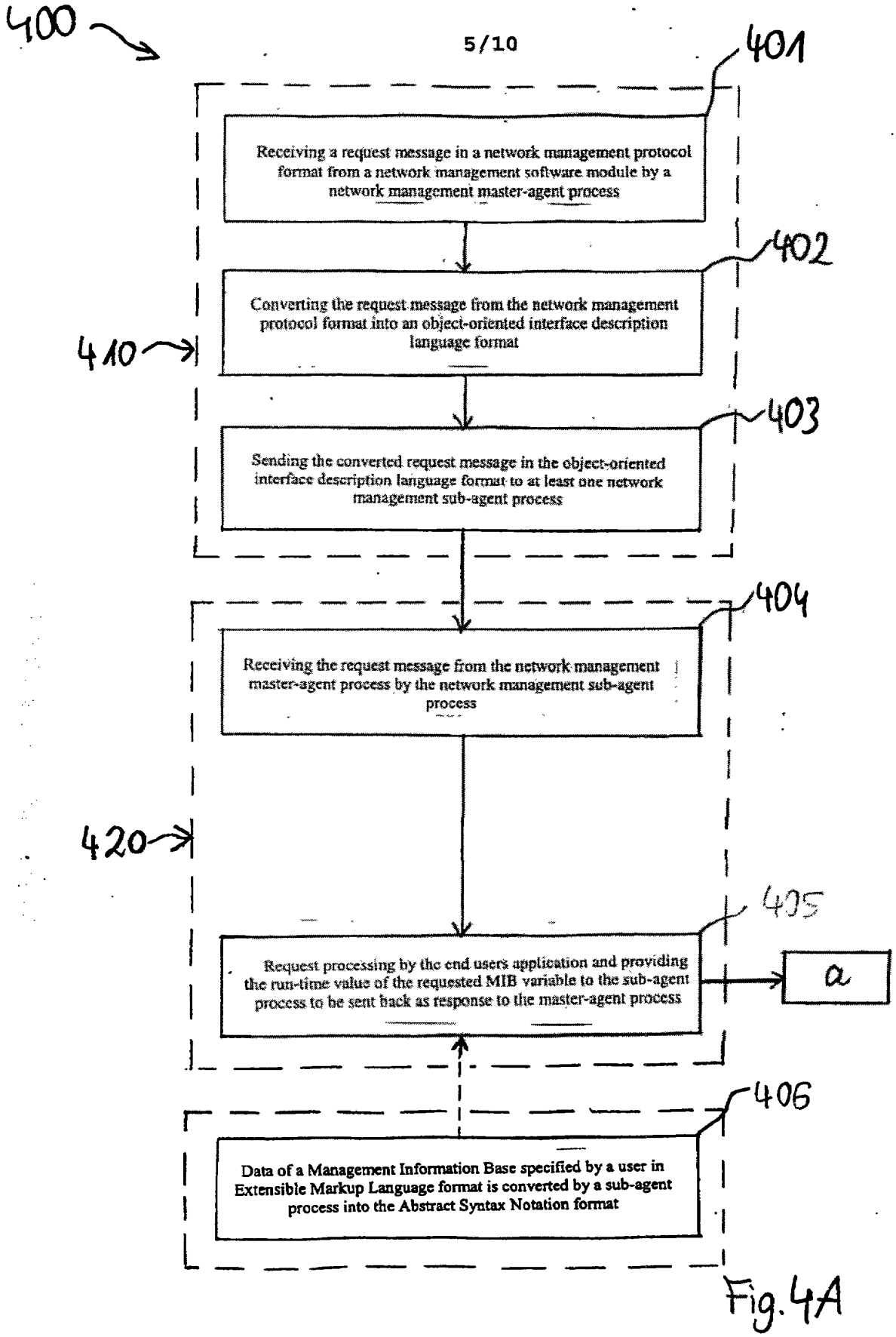

400
401
Receiving a request message in a network management protocol format from a network management software module by a network management master-agent process
402
Converting the request message from the network management protocol format into an object-oriented interface description language format
410
403
Sending the converted request message in the object-oriented interface description language format to at least one network management sub-agent process
404
Receiving the request message from the network management master-agent process by the network management sub-agent process
420
405
Request processing by the end users application and providing the run-time value of the requested MIB variable to the sub-agent process to be sent back as response to the master-agent process
a
406
Data of a Management Information Base specified by a user in Extensible Markup Language format is converted by a sub-agent process into the Abstract Syntax Notation format

Fig.4A

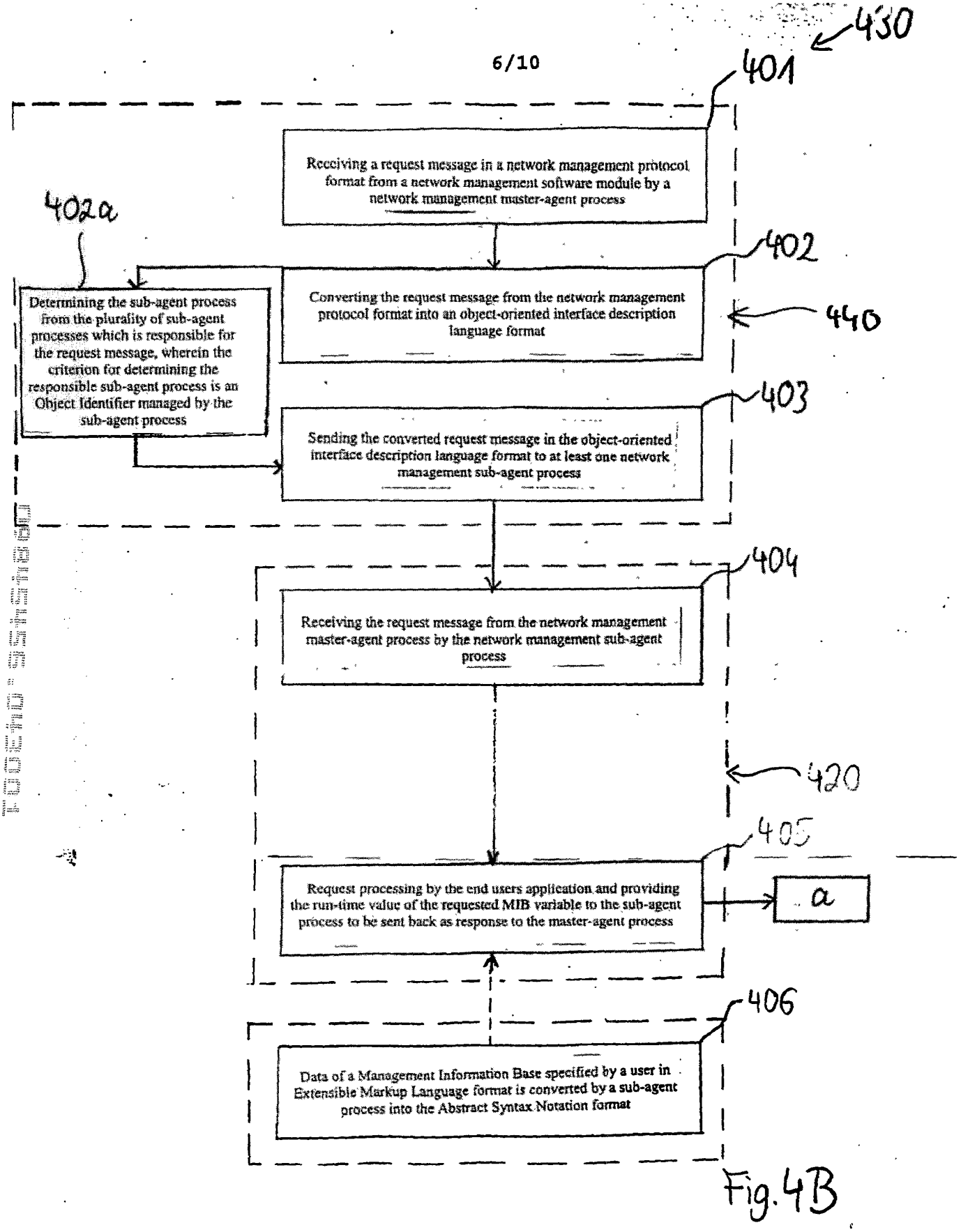

430
401
Receiving a request message in a network management protocol format from a network management software module by a network management master-agent process
402a
402
Converting the request message from the network management protocol format into an object-oriented interface description language format
440
Determining the sub-agent process from the plurality of sub-agent processes which is responsible for the request message, wherein the criterion for determining the responsible sub-agent process is an Object Identifier managed by the sub-agent process
403
Sending the converted request message in the object-oriented interface description language format to at least one network management sub-agent process
404
Receiving the request message from the network management master-agent process by the network management sub-agent process
420
405
Request processing by the end users application and providing the run-time value of the requested MIB variable to the sub-agent process to be sent back as response to the master-agent process
a
406
Data of a Management Information Base specified by a user in Extensible Markup Language format is converted by a sub-agent process into the Abstract Syntax Notation format

Fig.4B

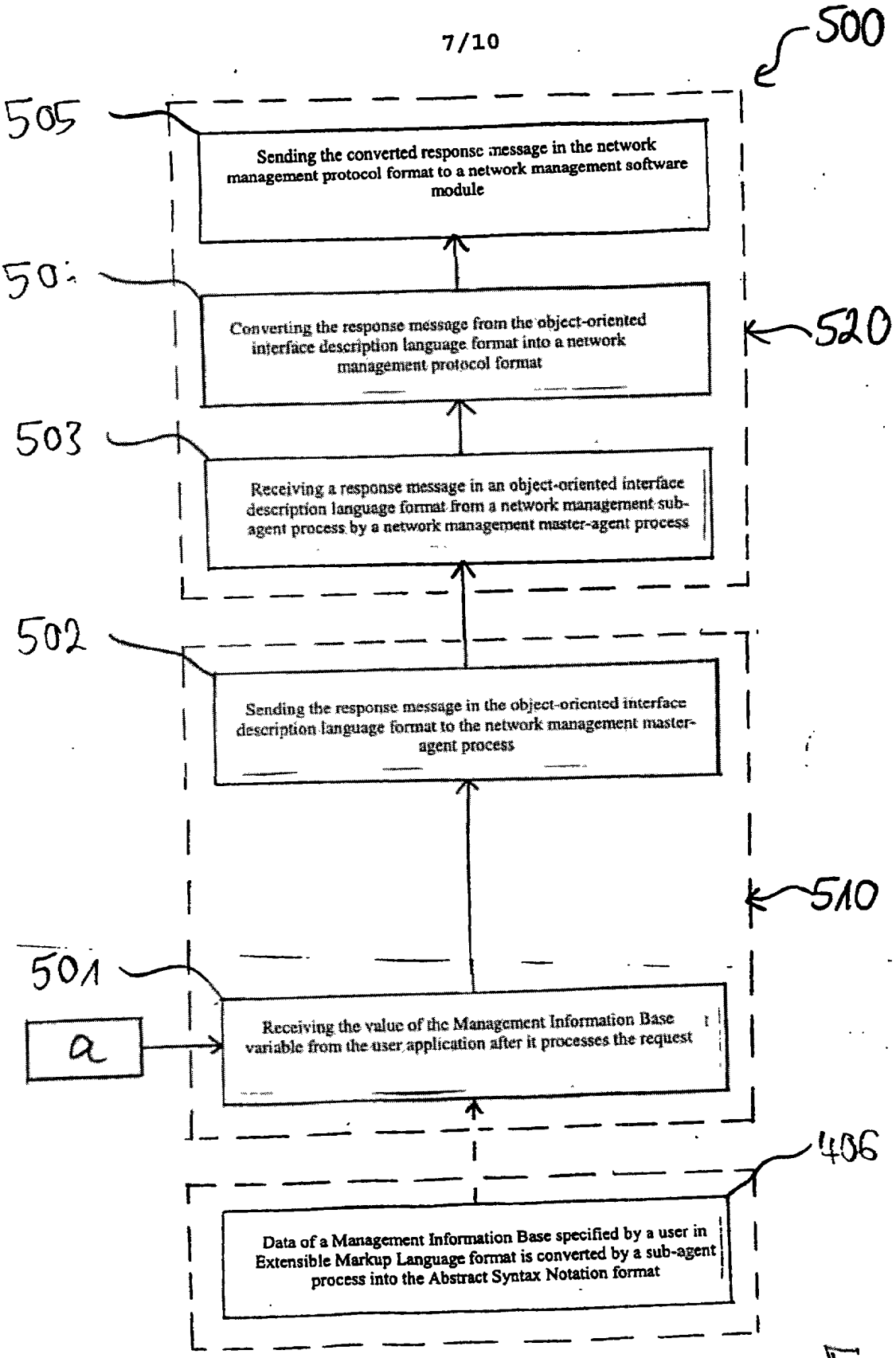

500
505
Sending the converted response message in the network management protocol format to a network management software module
504
Converting the response message from the object-oriented interface description language format into a network management protocol format
520
503
Receiving a response message in an object-oriented interface description language format from a network management sub-agent process by a network management master-agent process
502
Sending the response message in the object-oriented interface description language format to the network management master-agent process
510
501
a
Receiving the value of the Management Information Base variable from the user application after it processes the request
406
Data of a Management Information Base specified by a user in Extensible Markup Language format is converted by a sub-agent process into the Abstract Syntax Notation format

Fig.5

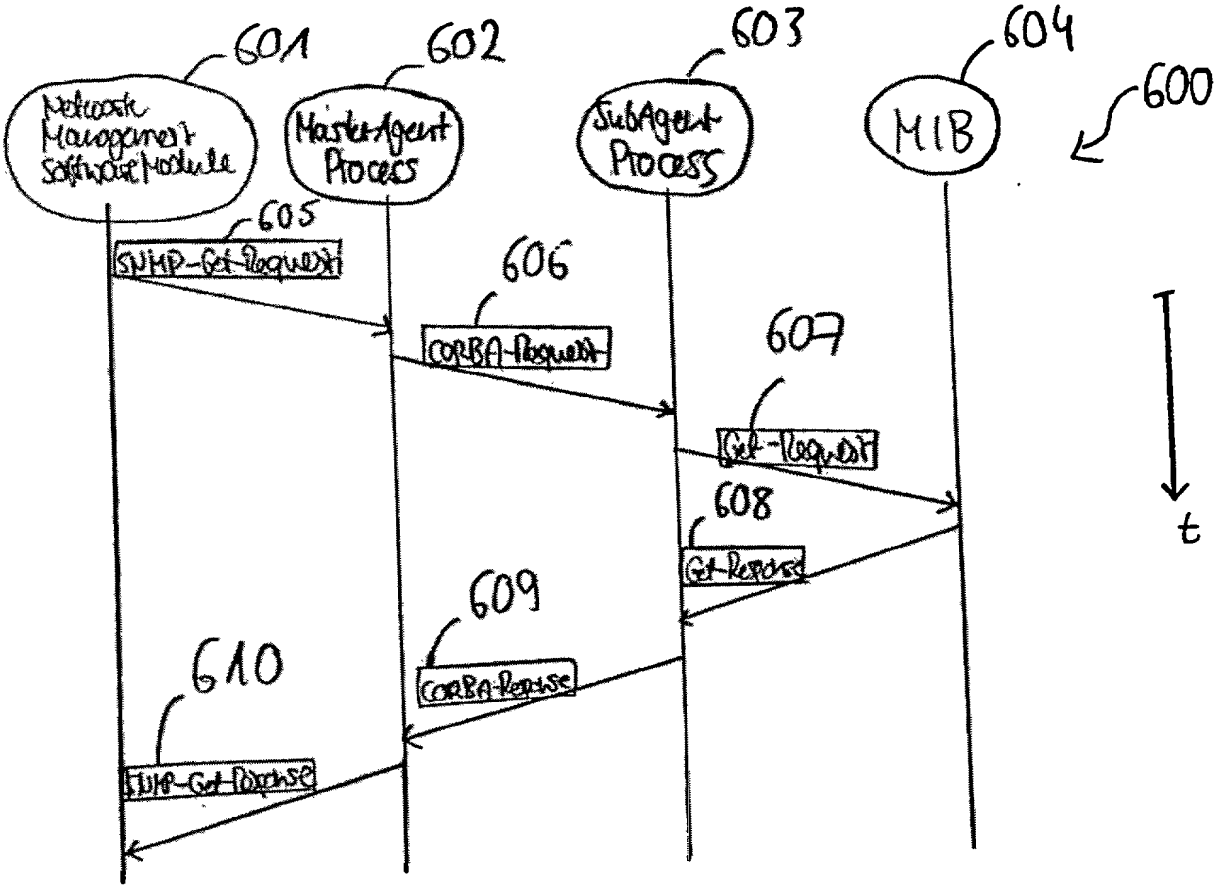

601
602
603
604
600
Network Management Software Module
Master Agent Process
SubAgent Process
MIB
605
SNMP-Get-Request
606
CORBA-Request
607
Get-Request
608
Get-Response
609
CORBA-Response
610
SNMP-Get-Response
t

Fig. 6

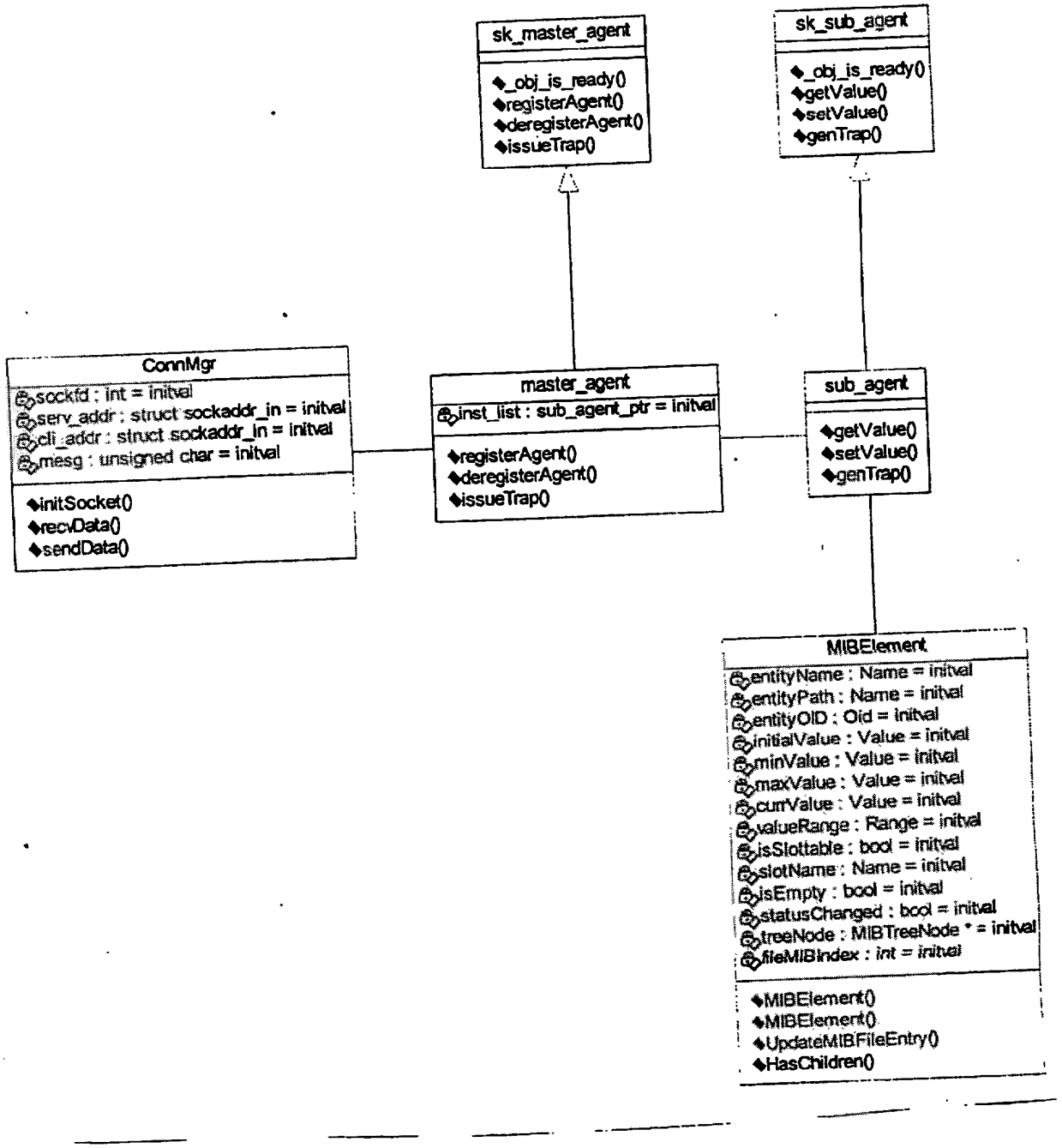
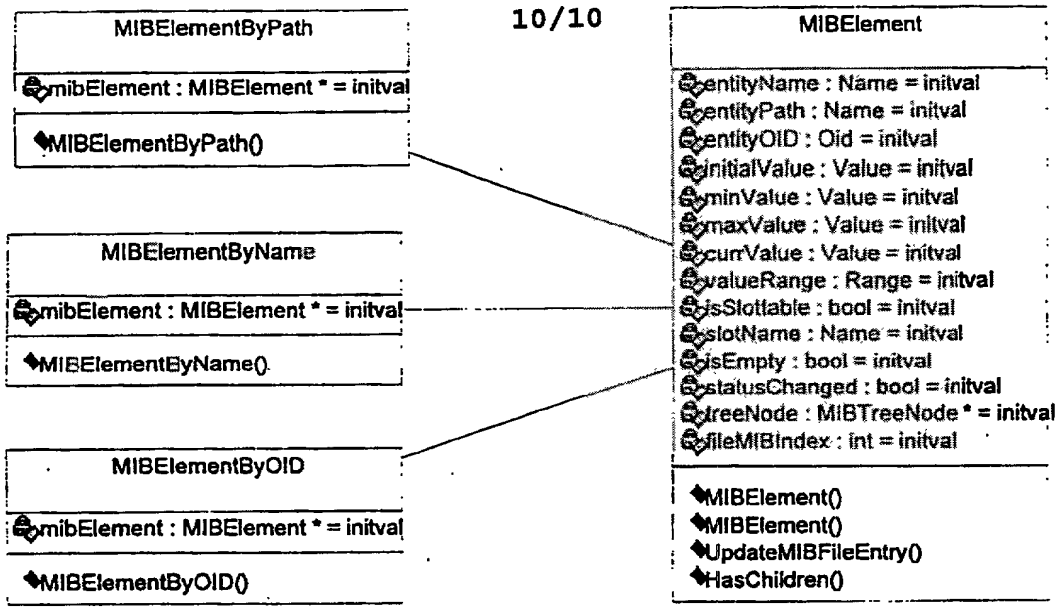
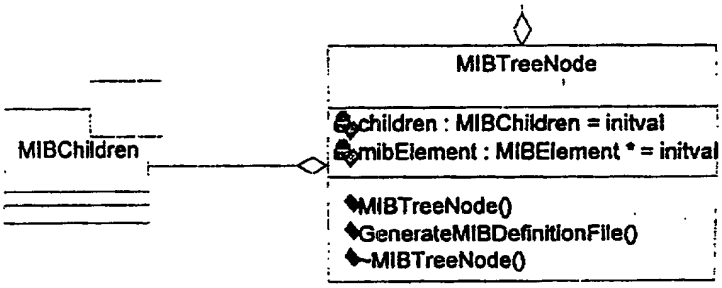

sk_master_agent
♦_obj_is_ready()
♦registerAgent()
♦deregisterAgent()
♦issueTrap()
sk_sub_agent
♦_obj_is_ready()
♦getValue()
♦setValue()
♦genTrap()
ConnMgr
sockfd : int = initval
serv_addr : struct sockaddr_in = initval
cli_addr : struct sockaddr_in = initval
mesg : unsigned char = initval
♦initSocket()
♦recvData()
♦sendData()
master_agent
inst_list : sub_agent_ptr = initval
♦registerAgent()
♦deregisterAgent()
♦issueTrap()
sub_agent
♦getValue()
♦setValue()
♦genTrap()
MIBElement
entityName : Name = initval
entityPath : Name = initval
entityOID : Oid = initval
initialValue : Value = initval
minValue : Value = initval
maxValue : Value = initval
curValue : Value = initval
valueRange : Range = initval
isSlottable : bool = initval
slotName : Name = initval
isEmpty : bool = initval
statusChanged : bool = initval
treeNode : MIBTreeNode * = initval
fileMIBIndex : int = initval
♦MIBElement()
♦MIBElement()
♦UpdateMIBFileEntry()
♦HasChildren()

Fig. 7


MIBElementByPath
mibElement : MIBElement * = initval
MIBElementByPath()
MIBElementByName
mibElement : MIBElement * = initval
MIBElementByName()
MIBElementByOID
mibElement : MIBElement * = initval
MIBElementByOID()
MIBElement
entityName : Name = initval
entityPath : Name = initval
entityOID : Oid = initval
initialValue : Value = initval
minValue : Value = initval
maxValue : Value = initval
curValue : Value = initval
valueRange : Range = initval
isSlottable : bool = initval
slotName : Name = initval
isEmpty : bool = initval
statusChanged : bool = initval
treeNode : MIBTreeNode * = initval
fileMIBIndex : int = initval
MIBElement()
MIBElement()
UpdateMIBFileEntry()
HasChildren()


MIBChildren
MIBTreeNode
children : MIBChildren = initval
mibElement : MIBElement * = initval
MIBTreeNode()
GenerateMIBDefinitionFile()
~MIBTreeNode()

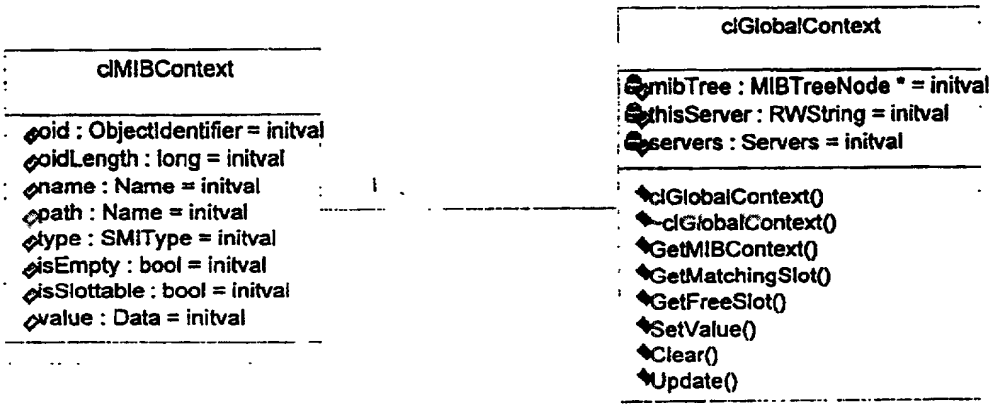

clMIBContext
oid : ObjectIdentifier = initval
oidLength : long = initval
name : Name = initval
path : Name = initval
type : SMIType = initval
isEmpty : bool = initval
isSlottable : bool = initval
value : Data = initval
clGlobalContext
mibTree : MIBTreeNode * = initval
thisServer : RWString = initval
servers : Servers = initval
clGlobalContext()
~clGlobalContext()
GetMIBContext()
GetMatchingSlot()
GetFreeSlot()
SetValue()
Clear()
Update()

Fig. 8